

Objektorientierte Modellierung unter Einsatz eines CASE-Tools im Informatikunterricht der Jahrgangsstufe 11

Stefan Moll
GI-Workshop
Bommerholz, 11.10.02

Rahmenbedingungen und Integrationsvoraussetzungen

- Anfangsunterricht der Jahrgangsstufe 11
- Wahl des objektorientierten Ansatzes
- Erfahrungen in der Schule mit „Stifte-und-Mäuse-Konzept“
- Programmiersprache: Objekt-Pascal in der Delphi-Programmierungsumgebung

Ausgangsfragen und -probleme

- (1) Wie kann die Vorstellung von verschiedenen Objekten und deren Zusammenwirken gefördert werden?
- (2) Wie können grafische Modellierungsmethoden besser genutzt werden?
- (3) Wie können die unterschiedlichen Ebenen der sach- und darstellungsbezogenen Modellierung klarer werden?
- (4) Wie kann der syntaktische Ballast der Programmiersprache für die Lernenden verringert werden?
- (5) Wie können die Bedeutungen einzelner Attribute und Methoden den Lernenden besser präsent sein?

Modellierung im IU

- Modellierungstechniken als Kern der Vermittlung von Modellierungskompetenzen
- Behutsame und einzelne Einführung
→ Beschränkung auf Grundkonzepte
- Erfahrungen mit grafischen Methoden der Ablaufmodellierung (z. B. Struktogramme), analog bei der Klassenmodellierung:
Oft geringer Nutzen für Modellentwicklung und -umsetzung → CASE-Tool
- Modellierungsbeispiele auf schmalen Grad zwischen „zu schwer“ und „zu leicht“

Problembereiche des objektorientierten Anfangsunterrichts

„Anwendungsproblem analysieren“

- ! Probleme beschreiben mit Hilfsmitteln / Verfahren, die man noch nicht kennt
- Erste Modellierungen zunächst vorgeben
- Anforderung: Modell verstehen
- CRC-Karten und Objektspiel
 - Anschauliches Nachempfinden des Modells
 - Bearbeitbarkeit des Modells (zu übertragen in UML)

Problembereiche des objektorientierten Anfangsunterrichts

„Objektorientierte Designprinzipien kennen lernen“

- ! MVC-Konzept verlockend, aber zu schwer
- Trennung von Sache und Darstellung in zeitlicher Trennung deutlich werden lassen:
 - Erst die Sache in der Analysephase,
 - dann die Darstellung in der Designphase

Problembereiche des objektorientierten Anfangsunterrichts

„Delphi lernen“

! Delphi ist m. E. für den objektorientierten Anfangsunterricht nicht zu empfehlen, da nicht strikt objektorientiert.

→ Zunächst Verwendung des „Stifte und Mäuse“-Pakets. (Auch mit Aufbau einzelner eigener Darstellungsklassen)

Problembereiche des objektorientierten Anfangsunterrichts

„Pascal lernen / Kontrollstrukturen“

! Umfangreiche Sprachsyntax

- Automatische Codegenerierung durch
 - Kleine Hilfsprogramme
 - CASE-Tool

! Methodenrealisierung erfordert Kontrollstrukturen

- Erste Erfahrungen mit Kontrollstrukturen vor Modellierungsprojekt
 - Einstiegssequenz aus „Stifte und Mäuse“, dabei
 - konsequentere objektorientierte Sichtweise in den Kleinprojekten
 - erste Klassenmodifikationen / -erweiterungen

Vorteile von CASE-Tools

- „Interaktive“ Erzeugung und Bearbeitung von Modellen
- Generierung von statischem Code (mit zum Teil komplexer Syntax)
- Begleitende Dokumentation, separate Ausgabe der Dokumentation und Integration in Programmcode

Zentrale Bewertungsfragen

- Ist die Darstellung sach- und adressatengerecht?
Entsprechen Möglichkeiten und Optionen dem zu vermittelnden Begriffsverständnis?

Attribut-Dialog in OTW 2.4

Selection of attributes

Selection | Settings | Extended settings | Documentation | Stereotypes

Name: vermoegen
Type: integer

Cardinality
Min: 0
Max: 1
Container: []

Visibility
 Public
 Protected
 Private

Extended
 Const
 Class attribute
Visibility Methods

Type of variable
 Value
 Pointer
 Reference

Role type
 Role
 Aggregation
 Composition

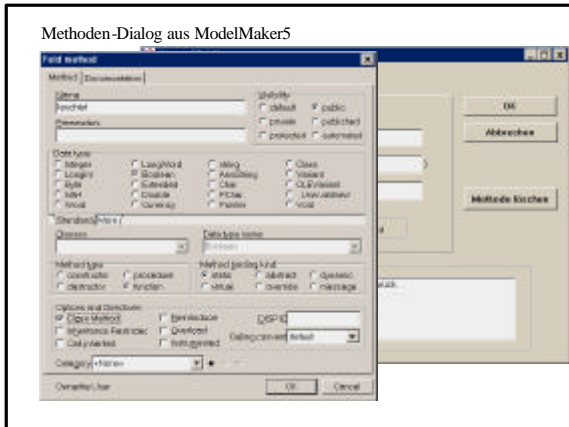
Context: Class attribute: Speicher

OK Abbrechen Übernehmen

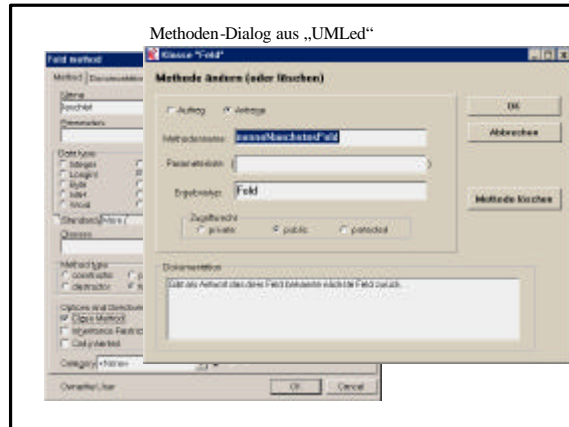
Zentrale Bewertungsfragen

- Ist die Darstellung sach- und adressatengerecht?
- Sind die notwendigen Bedienungsschritte überschaubar?

Methoden-Dialog aus ModelMaker5



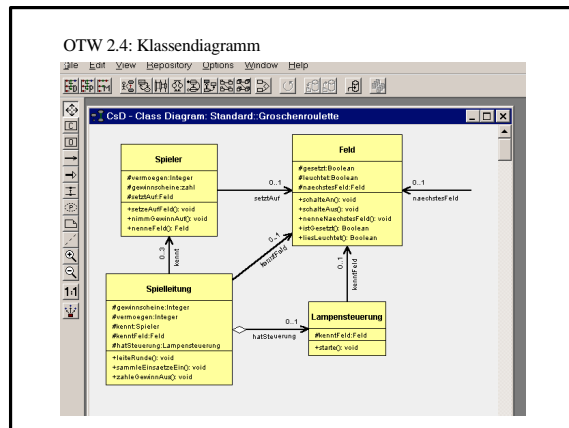
Methoden-Dialog aus „UMLED“



Zentrale Bewertungsfragen

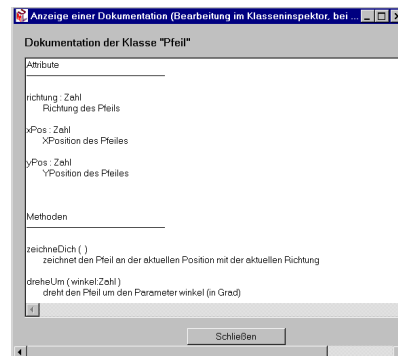
- Ist die Darstellung sach- und adressatengerecht?
- Sind die notwendigen Bedienungsschritte überschaubar?
- Können alle Informationen in kompakter, übersichtlicher Form angezeigt werden?

OTW 2.4: Klassendiagramm



Zentrale Bewertungsfragen

- Ist die Darstellung sach- und adressatengerecht?
- Sind die notwendigen Bedienungsschritte überschaubar?
- Können alle Informationen in kompakter, übersichtlicher Form angezeigt werden?
- Gibt es eine durchgängige Dokumentationsmöglichkeit mit einer kompakten Anzeige- und Speichermöglichkeit?



Zentrale Bewertungsfragen

- Ist die Darstellung sach- und adressatengerecht?
- Sind die notwendigen Bedienungsschritte überschaubar?
- Können alle Informationen in kompakter, übersichtlicher Form angezeigt werden?
- Gibt es eine durchgängige Dokumentationsmöglichkeit?
- Ist der automatisch erzeugte Code übersichtlich, strukturiert und compilierbar?

Vor dem ersten „Modellierungsprojekt“

- Arbeiten mit Objekten bzw. Klassen anhand „Stiften und Mäusen“:
Kleine Zeichenprogramme, wie Haus oder Ornamentband
- Projekt Pfeilwurf mit Klassen:
 - Klassen verwenden, Kontrollstrukturen erweitern
 - Klasse um Methode erweitern, UML-Editor nutzen.

Grobablauf

- OOA-Phase
 - Anwendungskontext erkunden, CRC-Modell erstellen
 - Modell in UML übertragen
 - Entwurf Benutzungsoberflächen
- OOD-Phase
 - Erweiterung um (technisch) notwendige Klassen, Attribute, Methoden, Beziehungen
- Implementationsphase
 - Code-Erzeugung mit dem Tool und Implementation

Das Beispiel: Groschenroulette



Das Beispiel: Groschenroulette



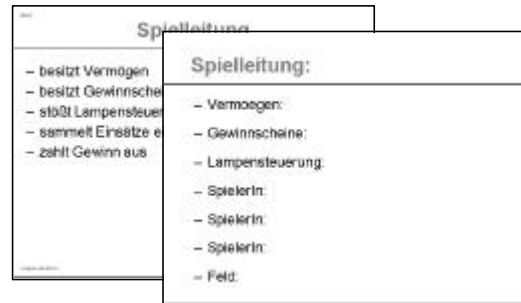
1. Schritt: Anwendungskontext erkunden (am Modell)



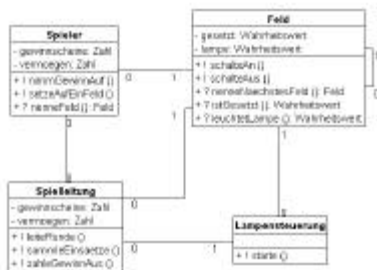
2. Schritt: CRC-Modell aufstellen (bzw. erkunden) und simulieren



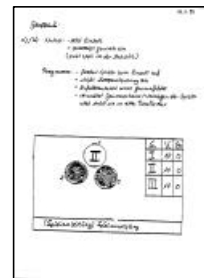
2. Schritt: CRC-Modell aufstellen (bzw. erkunden) und simulieren



3. Schritt: Übertragung in ein Klassendiagramm (UMLed)



4. Schritt: Entwurf der Benutzungsoberfläche



5. Schritt: Erweiterung des Modells

- Zunächst auf CRC-Ebene (Objektspiel nur kurz andeuten)
- Umsetzung in UML-Editor
- Eigene Modellierung von Darstellungsklassen (statt Rückgriff auf Delphiklassen)

6. bis 9. Schritt

- Differenzierung / Charakterisierung der Beziehungen
- Kriterien für eine gute Dokumentation (Reflexion der erstellten Dokumentationen)
- Zugriffsmethoden
- Konstruktoren / Destruktoren

10. Schritt: Implementation

- Codegenerierung durch UML-Editor als Basis
- Arbeitsteiliges Vorgehen
- Projektkommunikation mittels Austauschforum
- Arbeitsprozessdokumentation als Basis für folgenden Austausch (insb. über neue verwendete Konzepte)

Weitere Modellierungsprojekte

- Weniger Modellvorgaben, mehr Erarbeitung und Offenheit
- Einzelne Phasen kürzer oder ganz wegfallend
- Übergang zu Delphi-Umgebung nach der Behandlung von Aspekten wie Ereignisanwendung.